CLAIMS

What is claimed is:

1. An apparatus for determining a longest matching prefix, the apparatus comprising:

an external memory for storing one or more non-first-level tiny trees; and

an applications specific integrated circuit (ASIC) including a lookup engine and internal memory for storing a set of first-level tiny trees, the ASIC configured to perform operations, said operations including:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value;

retrieving a first level root node of said particular first-level tiny tree from said internal memory, the first-level root node including a first-level plurality of keys;

traversing said particular first-level tiny tree stored in said internal memory to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys, wherein the first-level tiny tree and the next-level tiny trees are independent trees;

retrieving a root node of the next-level tiny tree stored in the external memory, the root node including a plurality of keys to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree stored in the external memory to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

2. The apparatus of claim 1, comprising an associative memory, wherein said determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value includes performing a lookup operation on the associative memory based on the lookup value.

3. A method for identifying a longest matching prefix, the method comprising:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value;

retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys;

traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys, wherein the first-level tiny tree and the next-level tiny trees are independent trees;

retrieving a root node of the next-level tiny tree, the root node including a plurality of keys to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

4. The method of claim 3, the first-level tiny tree is associated with no back values.

5. A computer-readable medium containing computer-executable instructions for performing steps for identifying a longest matching prefix, said steps comprising:

determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value;

retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys;

traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys, wherein the first-level tiny tree and the next-level tiny trees are independent trees;

retrieving a root node of the next-level tiny tree, the root node including a plurality of keys to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys; and

identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

6. An apparatus for identifying a longest matching prefix, the method comprising:

means for determining which particular first-level tiny tree of a plurality of first-level tiny trees to search based on a lookup value;

means for retrieving a first-level root node of said particular first-level tiny tree, the first-level root node including a first-level plurality of keys;

means for traversing said particular first-level tiny tree to identify a next-level tiny tree, said traversing said particular first-level tiny tree including comparing the lookup value with one or more of the first-level plurality of keys, wherein the first-level tiny tree and the next-level tiny trees are independent trees;

means for retrieving a root node of the next-level tiny tree, the root node including a plurality of keys to compare with the lookup value and a back value to identify a matching prefix should no matching prefix be identified within said particular tree;

means for traversing said particular next-level tiny tree to either identify a matching prefix or a no match condition, said traversing said particular next-level tiny tree including comparing the lookup value with one or more of the plurality of keys; and

means for identifying as the longest matching prefix a prefix identified based on the back value if said traversing resulted in said no match condition else the matching prefix.

7. One or more data structures stored on one or more computer-readable medium, said one or more data structures comprising:

a hierarchy of a plurality of tiny trees for performing binary or multicolumn searches, the plurality of tiny trees including: a current-level tiny tree; and a next-level tiny tree; wherein the current-level tiny tree references the second-level tiny tree for one or more possible lookup values; and wherein the next-level tiny tree includes a back value for identifying a matching range whose match value is not represented within the next-level tiny tree; and

a plurality of indirect back values;

wherein the back value references a particular indirect back value of plurality of indirect back values, and at least one other indirect back value of the plurality of indirect back references a same result as the particular indirect back value.

8. The one or more data structures of claim 7, wherein for each particular tiny tree of the plurality of tiny trees, at least the first *m* bits of the matching value of each entry within said each particular tiny tree are the same, and *m* is at least four.

9. The one or more data structures of claim 8, wherein *m* is at least twelve.

10. A method for adding a prefix having a first endpoint and a second endpoint to a hierarchy of a plurality of tiny trees, the method comprising:

identifying that the first endpoint resides in a first tiny tree of the plurality of tiny trees and the second endpoint resides in a second tiny tree of the plurality of tiny trees;

updating the first tiny tree to identify the prefix for one or more first portions in the first tiny tree corresponding to where there are no prefixes longer than the prefix;

updating the second tiny tree to identify the prefix for one or more second portions in the second tiny tree corresponding to where there are no prefixes longer than the prefix;

updating a first back value corresponding to the first tiny tree to reflect the prefix; and

updating a second back value corresponding to the second tiny tree to reflect the prefix.

11. The method of claim 10, wherein the first and second back values reference a same result value corresponding to the prefix.

12. The method of claim 10, wherein the first and second back values reference different result values corresponding to the prefix.

13. The method of claim 10, wherein said updating the first tiny tree includes splitting the first tiny tree into the first tiny tree and a third tiny tree.

14. The method of claim 13, wherein a third back value is associated with the third tiny tree, and the value of the third back value is the value of the first back value prior to performing said updating the first back value.

15. A computer-readable medium containing computer-executable instructions for performing steps for adding a prefix having a first endpoint and a second endpoint to a hierarchy of a plurality of tiny trees, said steps comprising:

identifying that the first endpoint resides in a first tiny tree of the plurality of tiny trees and the second endpoint resides in a second tiny tree of the plurality of tiny trees;

updating the first tiny tree to identify the prefix for one or more first portions in the first tiny tree corresponding to where there are no prefixes longer than the prefix;

updating the second tiny tree to identify the prefix for one or more second portions in the second tiny tree corresponding to where there are no prefixes longer than the prefix;

updating a first back value corresponding to the first tiny tree to reflect the prefix; and

updating a second back value corresponding to the second tiny tree to reflect the prefix.

16. The computer-readable medium of claim 15, wherein the first and second back values reference a same result value corresponding to the prefix.

17. The computer-readable medium of claim 15, wherein the first and second back values reference different result values corresponding to the prefix.

18. The computer-readable medium of claim 15, wherein said updating the first tiny tree includes splitting the first tiny tree into the first tiny tree and a third tiny tree.

19. The computer-readable medium of claim 18, wherein a third back value is associated with the third tiny tree, and the value of the third back value is the value of the first back value prior to performing said updating the first back value.

20. An apparatus for adding a prefix having a first endpoint and a second endpoint to a hierarchy of a plurality of tiny trees, the apparatus comprising:

means for identifying that the first endpoint resides in a first tiny tree of the plurality of tiny trees and the second endpoint resides in a second tiny tree of the plurality of tiny trees;

means for updating the first tiny tree to identify the prefix for one or more first portions in the first tiny tree corresponding to where there are no prefixes longer than the prefix;

means for updating the second tiny tree to identify the prefix for one or more second portions in the second tiny tree corresponding to where there are no prefixes longer than the prefix;

means for updating a first back value corresponding to the first tiny tree to reflect the prefix; and

means for updating a second back value corresponding to the second tiny tree to reflect the prefix.

21. The apparatus of claim 20, wherein the first and second back values reference a same result value corresponding to the prefix.

22. The apparatus of claim 20, wherein the first and second back values reference different result values corresponding to the prefix.

23. The apparatus of claim 20, wherein said means for updating the first tiny tree includes means for splitting the first tiny tree into the first tiny tree and a third tiny tree.